

Jurnal Rekursif, Vol. 6 No. 1 Maret 2018, ISSN 2303-0755
<http://ejournal.unib.ac.id/index.php/rekursif/>

KAMUS KOSAKATA *BRITISH* *ENGLISH – AMERICAN ENGLISH* DENGAN SISTEM PENCARIAN FUNGSI HASH *MURMURHASH3* BERBASIS ANDROID

Muhammad Ikramullah¹, Ernawati², Aan Erlansari³

^{1,2,3}Program Studi Informatika, Fakultas Teknik, Universitas Bengkulu.
 Jl. WR. Supratman Kandang Limun Bengkulu 38371A INDONESIA
 (telp: 0736-341022; fax: 0736-341022)

¹ikram.watanabe@gmail.com

²ernawati@unib.ac.id

³aan_erlanshari@unib.ac.id

Abstrak : Bahasa Inggris telah berkembang dan membuat banyak variasi, diantaranya adalah *British English* dan *American English*. Kedua bahasa ini penting dan mempunyai banyak perbedaan, tetapi buku kamusnya cukup sulit ditemui. Tujuan dari penelitian ini yaitu membantu memudahkan pengguna dalam mencari kosakata menggunakan media kamus *British English – American English* yang tidak hanya praktis, tetapi juga lebih cepat. Sehingga dipilihlah *MurmurHash3* sebagai metode pencarian yang diimplementasikan pada penelitian ini dan berbasis *mobile*, dengan Android sebagai sistem operasinya. *MurmurHash3* merupakan sebuah fungsi *hash non-cryptographic* yang digunakan untuk sistem pencarian *hash*. *MurmurHash3* digunakan karena sifatnya yang stabil, cepat, sederhana, performa yang bagus dan *collision resistance* yang baik. Aplikasi ini dibangun menggunakan bahasa pemrograman Java dengan menggunakan IDE Android Studio 2.3.3 dan database SQLite. Pengujian yang dilakukan adalah *black box performance testing*, uji penelitian aplikasi terkait dan uji kelayakan sistem menggunakan kuesioner. Pengujian *black box performance testing* digunakan untuk menguji performa dari *MurmurHash3* dengan hasil pengujian sebesar 89.4%.

Kata Kunci : *MurmurHash3*, Android, Kamus, *British English*, *American English*.

Abstract: *English develops constantly and implemented as the search method of the research, produces many variations, such as the well known with Android as the operating system. British English and American English. Both of MurmurHash3 is a non-cryptographic hash these unique languages are important and have so function used for hash search systems. The many differences, unfortunately the dictionary is MurmurHash3 is used because of its stable, fast, quite difficult to find. The purpose of the research simple, great performance and good collision is to help users who is, in this case, looking for the resistance. This application is built using Java vocabulary of British English and American programming language by using IDE Android English differences with dictionary media which is Studio 2.3.3 and SQLite database. The tests not only practical to use, but also faster in terms of performed for the research are black box searching. The MurmurHash3 is choosen and to be performance testing, related application test, and*

system feasibility test with a questionnaire. The black box performance testing is used to test the performance of the MurmurHash3 search speed with 89.4% test results.

Keywords: MurmurHash3, Android, Dictionary, British English, American English.

I. PENDAHULUAN

Pentingnya *British English* dan *American English* harus didukung dengan ketersediaan sumber informasi berupa buku pembelajaran dan kamus kosakata yang mudah untuk ditemukan. Faktanya, setelah dilakukan peninjauan ke Gramedia sebagai toko buku terbesar di Bengkulu tidak ditemukan satupun buku pembelajaran maupun kamus *British English* dan *American English*.

Berkat adanya sistem operasi Android, para pengembang aplikasi *mobile* mulai membuat aplikasi pembelajaran *British English* maupun *American English*, mempermudah para pengguna Android untuk belajar kedua bahasa ini dan tersedia di *PlayStore*. *British English* dan *American English* adalah dua bahasa penting tetapi kamus dalam bentuk fisiknya masih sulit ditemui, sedangkan aplikasi kamus versi Android juga belum tersedia.

Metode yang digunakan pada penelitian ini adalah fungsi *hash MurmurHash3*. *MurmurHash* merupakan sebuah fungsi *hash non-cryptographic* yang digunakan untuk pencarian *hash*. Diciptakan oleh Austin Appleby yang versi terbarunya hingga penelitian ini diajukan adalah versi ketiga, yaitu *MurmurHash3*. *MurmurHash3* mempunyai beberapa kelebihan, diantaranya adalah proses yang cepat, sederhana, *collision resistance* yang

baik, serta performa yang bagus. *MurmurHash3* digunakan sebagai fungsi *hash* pencarian kata.

Demi mempermudah ketersediaan informasi kamus kosakata *British English* dan *American English*, penelitian ini mencoba memberikan solusi “Kamus Kosakata *British English – American English* berbasis Android”.

II. LANDASAN TEORI

A. Kamus

Kamus adalah sejenis buku rujukan yang menerangkan makna kata-kata dan berfungsi untuk membantu seseorang mengenal perkataan baru.

Adapun kamus dalam bentuk elektronik, kamus dalam bentuk buku memiliki kelebihan dalam jumlah kosakata yang banyak, tetapi juga memiliki kelemahan dalam hal pencarian kosa kata dan arti kata yang membutuhkan waktu yang lama [1].

B. *MurmurHash3*

MurmurHash adalah sebuah fungsi *hash non-cryptographic* yang digunakan untuk pencarian berbasis *hash*. *MurmurHash* diciptakan oleh Austin A pada tahun 2008. *MurmurHash* muncul dengan sejumlah versi, versi terbarunya adalah *MurmurHash3* yang menghasilkan 32-bit atau 128-bit nilai *hash* dan telah dirilis ke domain publik. *MurmurHash3* merupakan suksesor dan pengembangan dari versi sebelumnya, *MurmurHash2*. Nama *Murmur* berasal dari dua operasi dasar, *multiply* (MU) dan memutar atau *rotate* (R), yang digunakan berulang [2].

Proses *MurmurHash3* singkatnya seperti berikut:

```
k *= c1;  
k = rotl(k,r1);  
k *= c2;  
h^ = k;
```

$h_{rotr} = (h, r1);$

$h = h * m1 + n1;$

MurmurHash merupakan salah satu fungsi *hash non-cryptographic* tercepat yang dapat digunakan dan mempunyai distribusi ruang *hash* yang cukup seragam [3].

C. Android

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis Linux yang mencakup sistem operasi, *middleware*, dan aplikasi. Android merupakan sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan *platform* terbuka bagi para pengembang untuk membuat aplikasi mereka sendiri [4].

D. Database SQLite

SQLite merupakan paket perangkat lunak yang bersifat public domain yang menyediakan sistem manajemen basis data relasional atau RDBMS. Sistem basis data relasional digunakan untuk menyimpan *record* yang didefinisikan oleh pengguna pada ukuran tabel yang besar dan memproses perintah *query* yang kompleks dan menggabungkan data dari berbagai tabel untuk menghasilkan laporan dan rangkuman data [5].

III. METODE PENELITIAN

A. Jenis Penelitian

Jenis penelitian yang akan dilakukan adalah penelitian terapan (*applied research*). Penelitian ini akan menerapkan metode *MurmurHash3* sebagai salah satu fungsi *hash non-cryptographic* tercepat dengan kamus *American English – British English* berbasis Android sebagai objeknya. *MurmurHash3* merupakan versi terbaru dari fungsi *hash MurmurHash* hingga penelitian ini dilakukan. *MurmurHash3* sebagai fungsi *hash* akan menggantikan metode *string-matching*.

B. Data Set Penelitian

Data set yang digunakan dalam penelitian ini yaitu 2689 kosakata yang diperoleh dari buku “*The UK to USA Dictionary British English vs. American English*” edisi ketiga, oleh Claudine Dervaes dan John Hunter (2012).

C. Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan pada penelitian ini adalah *waterfall*. Adapun rincian tahapan - tahapan model *waterfall* pada penelitian ini adalah sebagai berikut:

1. Analisis Kebutuhan

Analisis kebutuhan bertujuan untuk mengidentifikasi semua kebutuhan dan garis besar aplikasi yang akan dibuat. Adapun analisis kebutuhan sistem yang akan dibuat adalah sebagai berikut:

- Kebutuhan data masukan. Data masukan yang dibutuhkan dalam aplikasi ini adalah 2689 kosakata yang diperoleh dari buku “*The UK to USA Dictionary British English vs. American English*” edisi ketiga, oleh Claudine Dervaes dan John Hunter.
- Kebutuhan data keluaran. Data keluaran yang dibutuhkan adalah data hasil transliterasi dari data kosakata yang berada di dalam *database*.
- Kebutuhan antarmuka. Kebutuhan antarmuka pada sistem adalah kemudahan pengguna dalam menggunakan aplikasi.

2. Desain

Tahap desain merupakan tahap konseptualisasi yang mengharuskan analisis dalam melakukan perancangan sistem yang akan dibangun. Diagram yang digunakan dalam perancangan sistem ini adalah *Unified Modelling Language* (UML).

3. Pembuatan Kode Program

Setelah menganalisa kebutuhan dan desain perancangan sistem maka akan ketahap selanjutnya yaitu pembuatan kode program dan implementasi. Implementasi merupakan tahapan nyata dalam penelitian ini.

4. Pengujian

Adapun pengujian sistem yang dilakukan di penelitian ini, dilakukan dengan pengujian fungsional pada sistem yang dibangun. Metode pengujian yang digunakan pada penelitian ini yaitu *black box testing*. Pada pengujian *black box* dibutuhkan informasi mengenai data masukan dan sasaran keluaran tanpa harus mengetahui bagaimana program atau sistem tersebut bekerja. Penelitian ini menggunakan pengujian *performance testings*.

Performance testings adalah teknik pengujian kecepatan suatu aplikasi dalam mengeksekusi suatu data yang telah di inputkan, yaitu dengan membuat tabel bank kosakata sebanyak 500 kosakata yang diambil secara acak. Kemudian di evaluasi dengan cara melakukan percobaan menjalankan aplikasi yang telah dibuat. Dalam hal ini, kecepatan metode pencarian menggunakan *MurmurHash3* akan dibandingkan dengan kecepatan metode pencarian menggunakan pencocokan nilai *string* Java Android (`string.equals(string)`).

5. Pendukung dan Pemeliharaan

Setelah aplikasi selesai dibangun, pengguna dapat menggunakan aplikasi. Jika terdapat pengembangan fungsional lebih lanjut dari aplikasi yang diinginkan oleh pengguna ataupun penambahan kosakata baru, maka akan dilakukan pemeliharaan.

D. Metode Uji Kelayakan Sistem

1. Kuesioner

Teknik pemilihan responden (sampel) dilakukan dengan metode *purposive sampling*, yaitu memilih sampel dari suatu populasi didasarkan pada informasi yang tersedia serta sesuai dengan penelitian yang sedang berjalan. Responden adalah mahasiswa – mahasiswi Universitas Bengkulu Fakultas Keguruan dan Ilmu Pendidikan Jurusan Bahasa Inggris berjumlah 20 orang, dengan rentang umur 19 – 24 tahun.

2. Tabulasi Data

Setelah mendapatkan data angket hasil kuesioner, dilakukan tabulasi data yaitu proses menempatkan data dalam bentuk tabel. Kemudian untuk proses perhitungan data angket menggunakan *skala likert*.

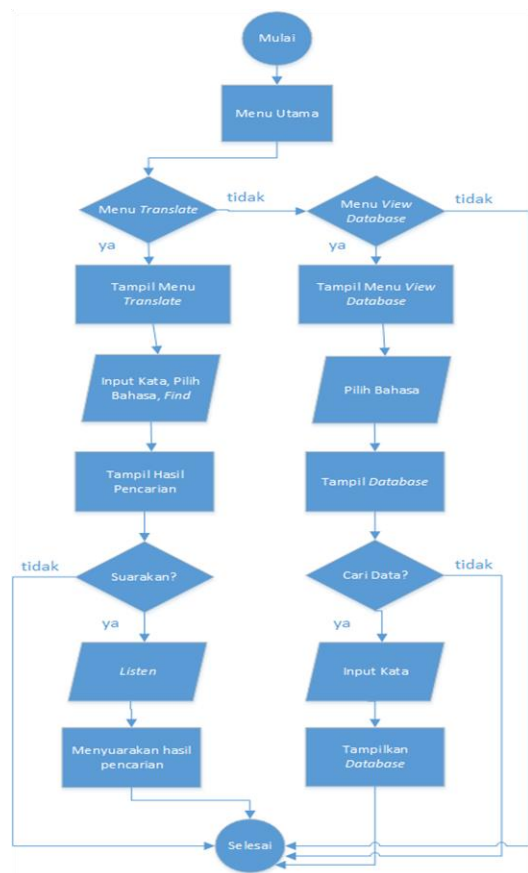
IV. ANALISIS DAN DESAIN PERANGKAT LUNAK

A. Analisis Sistem

Analisis sistem dilakukan dengan tujuan memperoleh informasi yang berhubungan dengan pembangunan aplikasi, mulai dari analisis antarmuka, alur kerja sistem, serta kebutuhan - kebutuhan non-fungsional. Keseluruhan informasi yang diperoleh akan dijadikan dasar pembangunan aplikasi Kamus Kosakata *British English – American English* Dengan Sistem Pencarian Fungsi *Hash Murmurhash3* Berbasis Android.

B. Alur Kerja Sistem

Untuk memahami sistem kerja dari aplikasi Kamus Kosakata *British English – American English* Dengan Sistem Pencarian Fungsi *Hash Murmurhash3* Berbasis Android, perhatikan Gambar 4.1.



Gambar 4. 1 Diagram Alur Kerja Sistem

C. Analisis Fungsional

Analisis fungsional memaparkan fitur – fitur yang akan dimasukkan ke dalam sistem yang dibuat. Adapun fitur – fitur aplikasi *British English - American English Dictionary* dengan metode *MurmurHash3* berbasis Android adalah:

1. Aplikasi mampu melakukan pencarian lebih cepat dengan metode *hashing MurmurHash3* dibandingkan dengan pencarian menggunakan *string*, dibuktikan dengan hasil rekap proses pencarian selama pengujian.
2. Aplikasi mampu menerjemahkan total 2689 kosakata UK-US maupun US-UK.
3. Aplikasi ini mampu memprediksi kata yang ingin dicari oleh *user* setiap kali menginputkan huruf-per-hurufnya di kolom pencarian menu ‘Translate’.

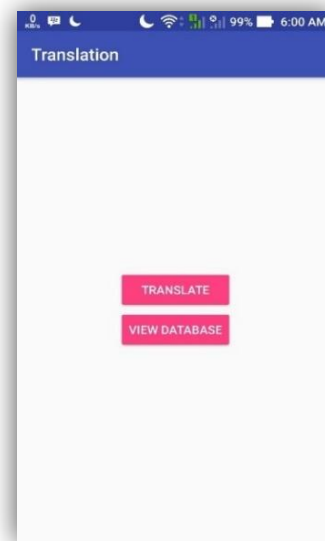
4. Aplikasi ini dapat menyuarakan hasil pencarian sesuai ejaan bahasa yang digunakan, baik ejaan *British* maupun ejaan *American*.

V. PEMBAHASAN

A. Implementasi Antarmuka

1. Tampilan Menu Utama

Screenshot Tampilan Menu Utama dapat dilihat pada Gambar 5.1.



Gambar 5.1 Tampilan Menu Utama

Pada Gambar 5.1 menunjukkan tampilan Menu Utama yang terdiri dari tombol menu *Translate* dan *View Database*. Tombol *Translate* merupakan menu pencarian kosakata (UK-US dan US-UK) menggunakan metode *MurmurHash3* yang merupakan tujuan dari dibangunnya aplikasi ini. Sedangkan tombol *View Database* adalah tempat melihat data – data spesifik yang ada di *database* seperti kosakata – kosakata berikut nilai *hash*-nya.

2. Tampilan Translate

Screenshot Tampilan *Translate* dapat dilihat pada Gambar 5.2.



Gambar 5.2 Tampilan Translate

Pada Gambar 5.2 *user* dapat melakukan pencarian dengan meng-inputkan kata. Pada Tampilan *Translate* terdapat kolom inputan kata yang ingin dicari, tombol pilihan bahasa 'UK-US' ataupun 'US-UK' untuk memilih bahasa translasi yang digunakan, tombol 'FIND' untuk memproses *hashing* dan melakukan pencarian kosakata, kolom hasil pencarian tempat menampilkan hasil pencarian, tombol 'LISTEN' jika *user* ingin mendengarkan ejaannya (ejaan *British* dan *American* berbeda), serta dibawahnya adalah kolom waktu kecepatan pencarian. Untuk mempermudah *user*, kolom pencarian menggunakan *AutoCompleteText* yang memunculkan prediksi kata yang ingin dicari oleh *user*.

Adapun proses *MurmurHash3* bekerja saat *user* menekan tombol 'FIND', yaitu saat sistem akan melakukan pencarian. Setelah didapatkan nilai *hash* dari algoritma *MurmurHash3* pada proses pencarian dilakukan pencocokan nilai *hash* dengan yang ada di *database*. *Screenshot* Tampilan Dengan Hasil Pencarian dapat dilihat pada gambar 5.3.

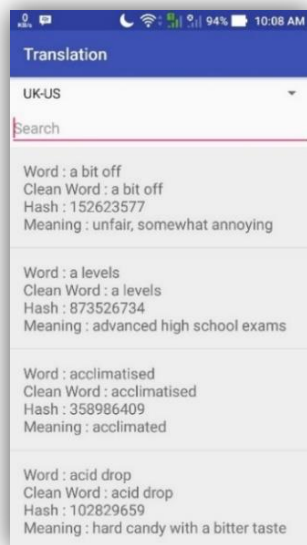


Gambar 5.3 Tampilan Translate Dengan Hasil Pencarian

Pada Gambar 5.3 menampilkan contoh dari hasil pencarian terjemahan kata. *User* meng-inputkan kata *football* lalu memilih bahasa translasi 'UK-US' dan menekan tombol 'FIND' maka hasil terjemahan *American*-nya adalah *soccer*. Kolom hasil kecepatan pencarian menggunakan format *millisecond* (ms). 'Preparation' pada kolom kecepatan pencarian adalah waktu proses *hashing MurmurHash3* terhadap kata yang diinputkan dan pembersihan kata dari simbol – simbol (jika ada). 'Searching' adalah waktu kecepatan proses pencarian dan pencocokan *hash* pada *database*. 'Total' adalah total waktu yang diperlukan untuk pencarian kosakata menggunakan *MurmurHash3*.

3. Tampilan View Database

Screenshot Tampilan *View Database* dapat dilihat pada Gambar 5.4.



Gambar 5.4 Tampilan View Database



Gambar 5.5 Tampilan View Database Dengan Contoh Clean Word

Pada Gambar 5.4 menampilkan menu *View Database*. Terdapat pilihan bahasa 'UK-US' dan 'US-UK' untuk melihat data pada masing – masing bahasa translasi. Dibawahnya terdapat kolom pencarian data jika *user* ingin mencari data kosakata tertentu. Pada kolom *database* terdapat 'Word' yang merupakan kosakata. Kemudian 'Clean' adalah *clean word* atau pembersihan kosakata dari simbol – simbol (contohnya *don't* menjadi *dont*) dan hanya sekedar membantu proses *hashing*, tidak berpengaruh ke hasil pencarian di menu *Translate*. Selanjutnya 'Hash' adalah *hash value* dari kosakata tersebut. Yang terakhir 'Meaning' merupakan arti kata. *Screenshot Tampilan View Database Dengan Contoh Clean Word* dapat dilihat pada gambar 5.5.

Pada Gambar 5.5 menunjukkan contoh fungsi 'Clean' dari simbol – simbol. Dapat kita lihat pada kata "*birthday honor's list*" yang dibersihkan menjadi "*birthday honors list*".

B. Pengujian Sistem Black Box Performance Testing

Pengujian *black box* sebagai pengujian pendukung untuk melihat kapabilitas dari aplikasi yang dihasilkan oleh penelitian ini. Pengujian *black box* yang digunakan pada penelitian ini adalah *Performance Testing* untuk mengukur akurasi kecepatan aplikasi dalam mengeksekusi kosakata yang diinputkan, yaitu dengan membuat tabel kosakata yang diambil secara acak. Kemudian di evaluasi dengan cara melakukan perbandingan percobaan menjalankan aplikasi yang telah dibuat yaitu pencarian menggunakan metode *MurmurHash3* dengan pencarian menggunakan *string-matching*. Dalam penelitian ini, kecepatan metode pencarian menggunakan *MurmurHash3* akan dibandingkan dengan kecepatan metode pencarian menggunakan pencocokan nilai *string* Java Android (*string.equals(string)*). Kemudian mencari persentase keberhasilan pengujian dengan acuan kecepatan pencarian tersebut.

Pengujian dilakukan dengan melihat waktu proses eksekusi kedua metode pencarian yang

dibandingkan. Semakin kecil nilai *output* maka semakin cepat proses pencarian kosakatanya. Jika *MurmurHash3* terbukti lebih cepat, pengujian BERHASIL. Jika *string-matching* lebih cepat, pengujian GAGAL. Pengujian menggunakan sampel 500 kosakata *British – American* maupun *American - British* yang diambil secara acak, yang beberapa diantaranya dapat dilihat pada Tabel 5.1.

Tabel 5.1 Pengujian Sistem *Black Box Performance Testing*

Adapun hasil pengujian didapatkan sebagai berikut:

Jumlah Data Sampel : 500 Kosakata
Pengujian Berhasil : 447
Pengujian Gagal : 53

No	Input		Output			Keterangan
	Kosakata	Arti Kata	<i>MurmurHash3</i> (ms)	String (ms)	Selisih Waktu	
1	acclimated	acclimated	0.02992	0.01802	0.0119	GAGAL
2	book	make reservations	0.02937	0.03079	0.00142	BERHASIL
3	petrol	gasoline, gas	0.08763	0.13328	0.04565	BERHASIL
4	dustman	garbage worker, sanitary engineer	0.05699	0.10062	0.04363	BERHASIL
5	turn ups	pant cuffs	0.12917	0.19527	0.0661	BERHASIL
I	I	I	I	I	I	I
500	hokey cokey	hokey pokey	0.0848	0.09825	0.01345	BERHASIL

Persentase Tingkat Akurasi Kecepatan:

Adapun rumus perhitungan persentase tingkat akurasi adalah:

$$\frac{\text{pengujian berhasil}}{\text{total sampel}} \times 100\%$$

Sehingga perhitungannya menjadi,

$$\frac{447}{500} \times 100\% = 89.4\%$$

Maka, tingkat akurasi pengujian *black box performance testing* pada penelitian ini adalah sebesar 89.4%.

Dari total 500 sampel kosakata yang dipilih secara acak, terbukti sebanyak 447 sampel

memiliki waktu proses pencarian lebih cepat menggunakan *MurmurHash3* dibandingkan *string-matching* yang hanya 53 sampel. Dengan tingkat akurasi kecepatan pencarian kosakata sebesar 89,4%, dapat disimpulkan bahwa *MurmurHash3* lebih akurat dibandingkan pencarian *string-matching*.

C. Pengujian Kelayakan Sistem

1. Kuesioner

Untuk kuesioner penelitian ini diberikan gradasi jawaban: SS = (Sangat Setuju); S = (Setuju); C = (Cukup); TS = (Tidak Setuju); STS = (Sangat Tidak Setuju). Dengan bobot penilaian jawaban adalah SS = 4; S = 3; TS = 2; STS = 1. Kemudian data dengan skala *Likert* dianalisis dengan menghitung skor akhir pada tiap interval dari pertanyaan yang diberikan ke responden. maka kategori penilaian yang dihasilkan dapat dilihat pada tabel 5.2.

Tabel 5.2 Interval dan Kategori Pilihan

Interval	Kategori
3.28 – 4.03	Sangat Setuju
2.52 – 3.27	Setuju
1.76 – 2.51	Tidak Setuju
1.00 – 1.75	Sangat Tidak Setuju

Kemudian data dengan skala *Likert* dianalisis dengan menghitung skor akhir pada tiap interval dari pertanyaan yang diberikan ke responden.

2. Hasil Uji Kelayakan Sistem

Hasil dari pengujian kelayakan sistem dengan variabel pengujian penggunaan aplikasi, pengujian kinerja aplikasi, dan pengujian fitur aplikasi menunjukkan bahwa hasil perhitungan ketiga variabel tersebut masing – masing berada dalam interval 3.28 – 4.03. Sehingga dapat disimpulkan penilaian dari 20 responden pada pengujian kelayakan sistem aplikasi ini termasuk dalam kategori “Sangat Setuju”.

D. Pengujian Penelitian Aplikasi Terkait yang Ada di PlayStore

Adapun pengujian penelitian aplikasi terkait yang ada di *PlayStore* adalah sebuah pengujian perbandingan fitur aplikasi hasil penelitian dengan aplikasi – aplikasi terkait yang ada di *PlayStore*, dalam hal ini fitur yang diuji yaitu kosakata *British English - American English*. Pengujian ini dilakukan dengan cara membandingkan kosakata yang diambil pada aplikasi yang dibangun dengan 7 aplikasi yang berkaitan dengan *British English - American English* sebagai bahan uji dan tersedia di *PlayStore*. Pengujian penelitian aplikasi terkait yang ada di *PlayStore* menggunakan kondisi sebagai berikut:

1. Jika pada kedua aplikasi ditemukan kosakata yang sama, maka pengujian **GAGAL**.
2. Jika pada kedua aplikasi tidak ditemukan kosakata yang sama, maka pengujian **BERHASIL**.
3. Jika aplikasi yang berkaitan dengan *British English - American English* sebagai bahan uji tidak ada fungsi translasi *British - American*, maka pengujian **BERHASIL** karena salah satu tujuan penelitian ini adalah membantu mengatasi keterbatasan kosakata pada aplikasi - aplikasi pembelajaran *British English & American English* yang tersedia di *PlayStore*.
4. Target keberhasilan pengujian ini minimal 50%.
5. Kosakata *British English* menggunakan aplikasi penelitian, dan kosakata *American English* sebagai pembanding menggunakan aplikasi uji.

Adapun hasil pengujiannya sebagai berikut:

Jumlah Data Sampel	: 7 Aplikasi
Pengujian Berhasil	: 5
Pengujian Gagal	: 2

Sehingga perhitungan akurasinya menjadi,
 $\frac{5}{7} \times 100\% = 71.4\%$

Dapat disimpulkan bahwa pengujian penelitian aplikasi terkait ini berhasil karena melebihi target awal 50%, yaitu sebesar 71.4%.

VI. PENUTUP

A. Kesimpulan

Berdasarkan dari penelitian yang telah dilakukan, terdapat beberapa kesimpulan yang diantaranya sebagai berikut:

1. Penelitian ini telah menghasilkan sebuah aplikasi *British English – American English Dictionary* dengan metode *MurmurHash3* berbasis Android.
2. Aplikasi telah menjalani pengujian *Blackbox Performance Testing* dengan membandingkan performa kecepatan pencarian *MurmurHash3* dengan *string-matching*. Hasil pengujian tersebut diketahui *MurmurHash3* lebih akurat dibandingkan *string-matching* dengan tingkat akurasi 89.4%.
3. Hasil dari pengujian kelayakan sistem dengan variabel pengujian penggunaan aplikasi, pengujian kinerja aplikasi, dan pengujian fitur aplikasi menunjukkan hasil perhitungan ketiga variabel tersebut masing – masing berada dalam interval 3.28 – 4.03. Sehingga dapat disimpulkan penilaian dari 20 responden pada pengujian kelayakan sistem aplikasi ini termasuk dalam kategori “Sangat Setuju”.
4. Aplikasi telah menjalani pengujian penelitian aplikasi terkait yang ada di *PlayStore* Hasil penelitian menunjukkan akurasi hasil pengujian sebesar 71.4% (berhasil).

B. Saran

Terdapat beberapa saran dalam pengembangan penelitian ini kedepannya, yaitu:

1. Diharapkan kedepannya dapat dikembangkan lagi dengan menerjemahkan kalimat.
2. Diharapkan pengembangannya dapat berjalan *offline* agar dapat digunakan dimanapun dan kapanpun dibutuhkan.
3. Diharapkan kedepannya bisa dilakukan penambahan jumlah kosakata *British English* maupun *American English*.

REFERENSI

- [1] Balbeid, A. (2008). *Rancang Bangun Aplikasi Mobile Kamus Dinamis Menggunakan Teknologi J2ME*. Diakses 5 Oktober 2016, dari <http://digilib.stikom.edu/detil.php?id=792>.
- [2] Appleby, A. (2012). *SMHasher & MurmurHash*. Diakses 4 Desember 2017 <https://scholar.google.com/scholar?q=Appleby%20A%20%282012%29%20SMHasher%20%26%20MurmurHash>.
- [3] Brown, C., & Junior, L. (2014). Efficient Cardinality estimation for k-mers in large DNA sequencing data sets. *FI000Research*, 1-5. doi:<http://dx.doi.org/10.1101/056846>.
- [4] Safaat H, N. (2012). *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC berbasis Android*. Bandung: Informatika.
- [5] Kreibich, J. A. (2010). *Using SQLite* (First ed.). (M. Loukides, Ed.) Sebastopol, CA: O'Reilly Media, Inc.